

R/D COMPUTING NEWSLETTER

CRITICAL FACTORS

During the past months, we've seen a great deal of criticism leveled at the companies that support our computer. With the status of "orphan" the only way any company engaged in supporting the 99/4A can survive is by keeping the information out in front of R & D and/or delivery dates **AND** sales! If you plan on continuing to use your machine, support your supporters. Otherwise we all might as well go buy Apples or IBM machines.

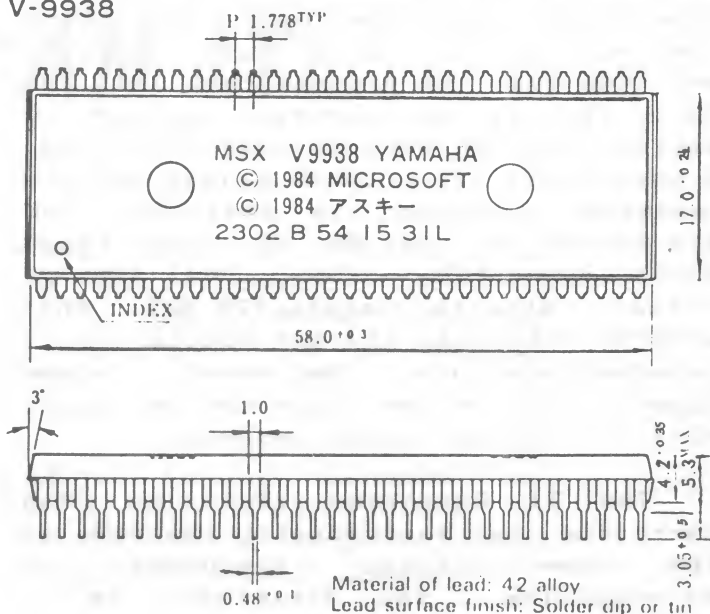
This reminds me of the time-honoured phrases: "Those that can, do. Those that can't teach." "ANYONE can be a critic." and "Opinion is the lowest form of wisdom."

Better yet, some of the comments (which include our own lack of timely delivery), border on the fringes of libel and severe misinformation. It is far better to yell at (complain, gripe or berate) those with whom you have a problem than to deal in rumour and innuendo which only serves to kill off the **ONLY** support you have left. Some individuals have stated that the Myarc machine will never be available in any commercial quantity. I believe otherwise. Others take shots at everyone out here - except for a personal favourite or two!

From experience, even the **BEST** and **BIGGEST** have glitches in producing and/or delivering products... Texas Instruments, Apple and IBM included. There is always another side to the story: failure of 'X' to deliver at one end or the other... chronic backlog... lack of support from the user community. One recent newsletter indicated that many new products do not reach the TI market due to lack of user support - which serves to create even

goto 5

V-9938



9938 Video Chip

One of the most exciting things about the Myarc machine (and the LONG awaited 80 column display unit) is the Yamaha V9938 Video Display Processor. This LSI (Large Scale Integration) chip (a 64 pin dual in-line package) is fully software compatible with the old 9918A contained in the 99/4A.

We have chip samples, spec books and programming guides here for the 9938. This is one hot chip.

Seven, yes seven, different modes! mouse & light pen inputs, up to 192k of Video RAM, 512 colours, superior bit map graphics, 512 x 400 pixel resolution and other interesting features: logical operations, up to 16 sprites per line, full buss speed operation etc.

Okay, what does all this mean? How does advanced **HIGH** resolution bit-mapped graphics sound? You can program each and every pixel (512 x 400) in a different colour. With the

goto10

TI EXPANSION BOX 8K RAM CARTRIDGE

by Ronald Zakariasen

The TI 99/4A cartridge port address' are from >6000 to >7FFF. Usually cartridges have ROM, GROM or EPROM located in this address space. However, if a RAM is placed in this address space, programs can be loaded from disk into the RAM. This means that you can copy cartridges to disk and then load the program back into the RAM. This is an indirect method of making a backup copy of your cartridge. A useful utility program called Machine Language Deloader is available for \$15.00 Cdn or \$10.00 US from Kings Marketing, P.O. Box 591 STN J, Calgary, Alberta, Canada T2A 4X8. This program requires the E/A module and it creates absolute compressed tagged object code files than can be loaded with E/A option 3 Load and Run.

The TI Expansion Box 8k RAM Cartridge is functionally the same as the Supercartridge discussed in Micropendium. The difference is the physical location of the RAM. Supercartridge has the RAM located in the E/A module, therefore the E/A module and the RAM are both connected to the computer simultaneously. This can also be accomplished by placing the RAM in the expansion box and the E/A module in the cartridge port. If the RAM is placed in the expansion box, it must be connected to the cartridge port addresses. Therefore in order to run a cartridge containing ROM or EPROM the RAM card must be disabled or removed. The switch SW1 on the circuit diagram disables the RAM and it can be located at the back of the expansion box. (For convenience, the switch can also be carried out to the front panel as well).

A description of the function of the IC 's used to build the RAM card is as follows: The 74LS245 IC is a bi-directional octal data bus driver. The memory IC is numbered 6264LP-15 and it is an 8k static CMOS RAM with an access time of 150 nanoseconds. The LP stands for low power which means the memory can be battery backed if desired. The

address line numbering for least significant bit in the memory is A0. This is backwards to the TI convention in which A15 is the least bit. The 74LS138 IC is a three to eight line decoder. When the address is between >6000 and >7FFF, the three most significant address bits are 011, therefore Y3 is selected and is used to enable the memory. The 74LS368 IC is an inverting bus driver with a tri-state output. This means that the output goes to a high resistance state when the IC is disabled. This is important because its output is used to enable the data bus drivers in the TI cable. High resistance is required to prevent interference with the other cards which must also enable these bus drivers. The 74LS368 IC is also used to invert DBIN in order to make it compatible with the memory output enable. It also inverts the memory enable signal so it can be used to control the transistor that drives the LED.

The components used to make the card are common and you should not have difficulty finding them. I used a Vector board but a printed circuit board would make construction easier. The connector in the expansion box is a 60 pin connector with .10 inch spacing between the contacts. If you don't already have a 32k memory expansion card, this design could be expanded to include an additional 32k. If you build and use a RAM cartridge, you do so at your own risk. Certain hardware additions [mistakes!] have the potential to damage other computer components.

Ronald Zakariasen January 1986.

The circuit diagram follows.



**Something for everyone
who didn't buy their TI-99/4A
so they could wait, and wait ,
and wait...
PRE-SCAN IT!**

Announcing a program for all those people that want fast performance from their Extended BASIC programs - and who doesn't?

What is Pre-Scan It!? Only the most remarkable utility for Extended BASIC users and programmers since the program compactor. Pre-Scan It!, simply enough, rewrites your TI, Myarc, Mechatronics, or Miller's Graphics modified Extended BASIC programs so that they not only execute fast, but nearly instantly!

At last! Now you will never again have to wait through those coffee break long pauses at the beginning of your favorite Extended BASIC programs. Finally, you can now write an Extended BASIC program any way you like, and then run it through Pre-Scan It! to get a program that starts up as fast as all those assembly and Forth programs!

If Pre-Scan It! only did this, it would be remarkable, but Asgard Software never stops at the merely remarkable. Pre-Scan It! will also make your program smaller in ways that only it's author J. Peter Hoddie, a master of Extended BASIC, knows. Pre-Scan It! will save bytes as well as time, and to top it off, it is compatible with all current versions of Extended BASIC, and all to come.

And finally, if all this isn't enough, Pre-Scan It! is available for the totally unbeatable price of only \$10.00! That's right, a single saw-buck will get this fantastic program delivered right to your door as fast as the U.S. Mail can carry it.

*Pre-Scan It! requires TI Extended BASIC and a disk drive system. A memory expansion unit is also required for some features, but is optional for most. Now available from official Asgard Software dealers, on CompuServe from TeleData*Guide (page TDG-4), and on The Source by writing via Source Mail to TI9720.*



Asgard Software

P.O. Box 10306

Rockville, MD 20850





RAMBO'S GUIDE TO REAL PROGRAMMERS

1. Real programmers don't write specs - users should consider themselves lucky to get any programs at all and take what they get.

2. Real programmers don't write application programs! they program right down to the bare metal. Application programming is for feebs who can't do systems programming.

3. Real programmers don't eat quiche. In fact, real programmers don't know how to spell quiche. They eat Twinkies and Szechwan food.

4. Real programmers don't write in COBOL. COBOL is for wimpy application programmers.

5. Real programmers' programs never work right the first time. But if you throw them on the machine they can be patched into working in "only a few" 30 hour debugging sessions.

6. Real programmers don't write in FORTRAN. FORTRAN is for pipe stress freaks and crystallography weenies.

7. Real programmers never work from 9 to 5. If any real programmers are around at 9 AM it's because they were up all night.

8. Real programmers don't write in BASIC. Actually, no programmers write in BASIC after the age of 12.

9. Real programmers don't play tennis, or any other sport that requires you to change clothes. Mountain climbing is OK, and real programmers wear their climbing boots to work in case a mountain should suddenly spring up in the middle of the machine room.

10. Real programmers don't write in PASCAL, BLISS or ADA, or any of those pinko computer science languages. Strong typing is for people with weak memories.

~The Canadian Pick Users Newsletter.

less company incentive - which creates less support. Cyclical. Consider the antique Apple II design... products are still being developed for the machine!

One example is a software company president that stated in effect, "we subscribe to a number of public database services - partly to let others know that we are watching the rumour mill. They are less critical when you're on line".

Another example: Mechatronic had also considered a user group demo program. When two other companies detailed the heavy losses sustained by engaging in this type of support operation the program idea was dropped.

Mechatronic will not send out sample hardware due to the potential of losing money. Not good - for their product exposure or for those who would like to buy new products. This in turn affects new software development to take advantage of the hardware coming out.

Chicago TI COMPUTER FAIRE 1986

The fourth annual premier TI Show in North America is taking place this year on Saturday, November 1st, 1986 sponsored by the Chicago TI 99/4A Users Group. The hours have been extended from 9:00 AM until 6:00 PM in the Ironwood room at Triton College. Triton is located at 2000 North Fifth Avenue in River Grove, Illinois.

The Chicago group has over 500 members and the attendance at the show is expected to reach 2000 people this year. Mass publicity in various publications and on 4A bulletin boards will serve to inform TI owners across North America. Hardware, software, information and new products will be shown by major 4A support companies.

We have reserved booth space as of this month. Come by to meet Ryte Data representatives in person. Several new products will be shown from ATRONIC and in-house developments. We will also be taking entries for the product draw to be held two weeks later. See you there!

C 99

The 'c' programming language is receiving a lot of attention - for good reason! Randy Holcomb stated that this development would contribute more to the survival of the 99/4A than anything else.

Many owners do seem to be confused by the nature of 'c'. Some people think that 'C' is "CPM"... not at all. C was developed at Bell Labs in the late 70's. Development progressed over a period of years starting way back in the 1960's. It was designed to support UNIX on DEC (Digital Equipment Corp.) mini computers. Most of the UNIX [multi-user, multi-tasking] operating system is written in 'C'. Today, many software houses use 'C' to design and implement application programs: dBase III, Wordstar 2000, Lotus 1-2-3 etc.

One thing that you should realize, 'c' is not a high level language like Basic, Logo or Pilot. It is closer to a cross between assembly and Pascal. It does make programming "close to the machine" easier and more powerful.

Another point important point is that 'c' is portable. You can code routines written for other machines for the TI 99/4A ie: programs written in 'c' for the IBM PC, the Atari ST or the Commodore AMIGA can be run on the TI and visa versa through the source code listing. This of course, is assuming that the other machine has its own version (usually \$200) of a 'c' compiler. Now compare that cost with the \$20 that Clint Pulley has suggested as Fair.

Many people have suggested that C99 is the most powerful language available for the TI to date. You can even include assembly language routines in your source file to be included for additional speed.

Among 'c's very powerful features, are the extensions which can be developed as "libraries" of routines. These libraries exist on disk to be called into the source code. These are designed to perform everything from graphics management to floating point mathematics. They are loaded and compiled at the same time as your

source code. Thus, you can have standard routines that can be used within many different programs.

The other attractive feature of 'c' is that it produces very fast, compact and efficient code. Some people have estimated that similar routines written in Pascal would be three or four times as large. You can imagine how this affects the speed of operation!

We have here three different 'c' example programs to demonstrate the syntax and use of various program statements. With luck, this should be a regular column: please submit programs to be included!

Final C note: We received the KERMIT listing mentioned last issue from Larry Butcher of Ohio. KERMIT is a standard communications protocol which allows different computers to transfer files fairly easily. To adapt this to the TI 99/4A will take extensive work.

SUPER CLOCK SUPPORT

For those of you who own a CorComp Clock Peripheral, Monty Schmidt has written a utility package that gives you more use and command of your clock. Either the Triple Tech card or the Stand-alone clock can be accessed more easily in Extended Basic with this program.

Direct access to the clock ROM at assembly speed gives you these features: three independent timers to set and read; alarm function; two interrupt routines to display time & date on screen with a keystroke (CTRL T) - continuously or on your command; all time & date displays are in 12 or 24 hour format using TEXT. This program also allows the week, date and time to be set independently - rather than all at once.

The uses in application programs and bulletin board systems are limitless. The disk is not copy protected to allow you full use in your Extended Basic programs. Package includes disk and manual. Only \$17.95 plus \$2.00 shipping (US funds).

```
/* Lines en C par Jean-Luc Brousseau */
/* type all in small letters, include
on disk random;c and bitrtn */
```

```
#include dsk1.random;c
#include dsk1.bitrtn
```

```
main()
{
    int ax1, ay1, ax2, ay2, cnt ;
    int x1, y1, x2, y2, colr ;
    bitmap(16,2) ;
    for (;;)
    {
        bitclr() ;
        ax1 = rnd (11) - 5 ;
        ay1 = rnd (11) - 5 ;
        ax2 = rnd (11) - 5 ;
        ay2 = rnd (11) - 5 ;
        x1 = rnd (226) + 15 ;
        x2 = rnd (226) + 15 ;
        y1 = rnd (162) + 15 ;
        y2 = rnd (162) + 15 ;
        for (cnt=0; cnt<150; cnt++)
        {
            colr = rnd (14) + 3 ;
            line(x1, y1, x2, y2, colr) ;
            x1 = x1 + ax1 ;
            y1 = y1 + ay1 ;
            x2 = x2 + ax2 ;
            y2 = y2 + ay2 ;
            if (x1 <= 7)
                ax1 = - ax1 ;
            if (y1 <= 7)
                ay1 = - ay1 ;
            if (x2 <= 7)
                ax2 = - ax2 ;
            if (y2 <= 7)
                ay2 = - ay2 ;
            if (x1 >= 248)
                ax1 = - ax1 ;
            if (x2 >= 248)
                ax2 = - ax2 ;
            if (y1 >= 184)
                ay1 = - ay1 ;
            if (y2 >= 184)
                ay2 = - ay2 ;
        }
        for (cnt=0; cnt<32000; cnt++)
        {
        }
    }
}
```

The c99 version should be created using the Editor Assembler Editor, and use the filename TWELVE;C to save the program in DIS/VAR 80 format. You will need the new version of c99, with the files PRINTF GRF1RF RANDOM;C and CONIO in disk drive 1 to be able to compile the C code. Use E/A option 5 to load C99C from disk 1. Answer Y to the first two

questions, and then give DSK2.TWELVE;C as the input filename and DSK2.TWELVE;S as the output filename. After compiling with no errors, load the E/A Assembler to assemble TWELVE;S with no options and the object file TWELVE;O. To run the program, use E/A #3 and load CSUP GRF1 PRINTF and TWELVE;O and then give START as the program name to start the program.

```
/* REM TWELVE IN C BY JON KELLER */
/* CONVERTED FROM AN EXTENDED BASIC */
/* PROGRAM BY WESLEY R RICHARDSON */
/* BLUEGRASS 99 COMPUTER SOCIETY INC. */
/* JUNE 1986 */
/*
```

```
#asm
    REF PRINTF
#endasm
#define one 1
#define none -1
#define sp 32
#include dsk1.conio
#include dsk1.grf1rf
```

```
main() /* MAIN PROGRAM */
{
    int dly,sidt,aodd,trs,cor,odd,bal;
    int wgt,sid,alp,ans,ava,bl,hl,sr,aga;
    clear();
    grf1();
    tab(9);
    printf("T W E L V E\n");
    printf("\n");
    printf(" THERE ARE TWELVE BLOCKS\n");
    printf(" MARKED A THROUGH L. ELEVEN\n");
    printf(" OF THESE HAVE THE SAME\n");
    printf(" WEIGHT. ONE BLOCK IS EITHER\n");
    printf(" HEAVIER OR LIGHTER THAN THE\n");
    printf(" OTHERS. THE TASK IS TO USE\n");
    printf(" THREE BALANCE WEIGHINGS TO\n");
    printf(" DETERMINE THE ODD ONE, AND\n");
    printf(" IF IT IS HEAVY OR LIGHT.\n");
    printf(" THE ARROW OVER THE BALANCE\n");
    printf(" INDICATES THE HEAVIER SIDE.\n");
    printf("\n");
    printf(" PRESS ANY KEY \n");
    printf("\n");
    wait();
    chrdef(91,"FFFF000000000000");
    chrdef(124,"FFFF183C66C38100");
    chrdef(93,"10387CFE10101010");
    chrdef(94,"F0E0E09008040201");
    chrdef(126,"0F07070910204080");
    /* rem [ | ] ^ ~ */
    trs=0;
    cor=0;
    rrandomize();
    wait();
    aga='Y';
    while(aga=='Y')
```

...TWELVE

```
{ odd=rnd(24);
odd=odd+1;
if(odd>12) odd=odd-25;
if(odd<0) aodd=-odd; else aodd=odd;
clear();
bal=0;
while(bal++<3)
{ wgt=0;
poscur(2,4);
printf("A B C D E F G H I J K L\n");
poscur(4+bal*4,3);
printf("[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[\n";
sid=None;
while(sid++<1)
{ alp=0;
while(alp++<6)
{ ava=32;
while(ava==32)
{ ans=accept(3+bal*4,
1+alp*2+sid*14,'A','L');
bl=0;
if(ans==10) bl=3;
if(bl==3) break;
ava=gchar(2,(ans-65)*2+4);
dly=0;
while(dly!=1000)
{ dly=dly+1;
}
}
if(bl==3) break;
hchar(2,(ans-65)*2+4,32,2);
wgt=wgt-2+sid*4;
if(ans-64==aodd)
wgt=wgt+(odd*(sid*2-1))/aodd;
}
}
bl=0;
if(wgt==0) hchar(3+bal*4,15,93,1);
if(wgt<0) hchar(3+bal*4,15,94,1);
if(wgt>0) hchar(3+bal*4,15,126,1);
}
poscur(18,3);
printf("ODD BLOCK (A-L) ?\n");
ans=10;
while(ans==10)
{ ans=accept(18,21,'A','L');
}
poscur(19,3);
printf("HEAVY OR LIGHT ?\n");
hl=10;
while(hl==10)
{ hl=accept(19,20,'H','L');
}
sr=0;
if(ans!=aodd+64) sr=1;
if(odd>0&hl==76) sr=1;
if(odd<0&hl==72) sr=1;
```

```

poscur(20,8);
if(sr!=1) printf("*** CORRECT ***");
if(sr==1) printf("*** SORRY ***");
    else cor=cor+1;
trs=trs+1;
poscur(22,3);
printf("correct=%d  Tries=%d",cor,trs);
poscur(24,3);
printf("Play Again (Y-N) ?");
aga=10;
while(aga==10)
{ aga=accept(24,22,'N','Z');
}
}
} /* END OF MAIN PROGRAM */

poscur(rw,cl)
    int rw,cl;
{    locate(rw,cl);
return;
}

tab(num)
int num;
{    while(num--!=0)
    {    printf(" ");
    }
return;
}

accept(row,col,lw,hw)
    int row,col,lw,hw;
{    int cha,st;
    cha=0;
    while(cha<lw|cha>hw)
    {    locate(row,col);
        cha=getchar();
        if(cha==10) break;
    }
return(cha);
}

wait()
{    int s,c;
    c=0;
    s=0;
    while(s==0)
        {s=poll(c);
        }
    return;
}

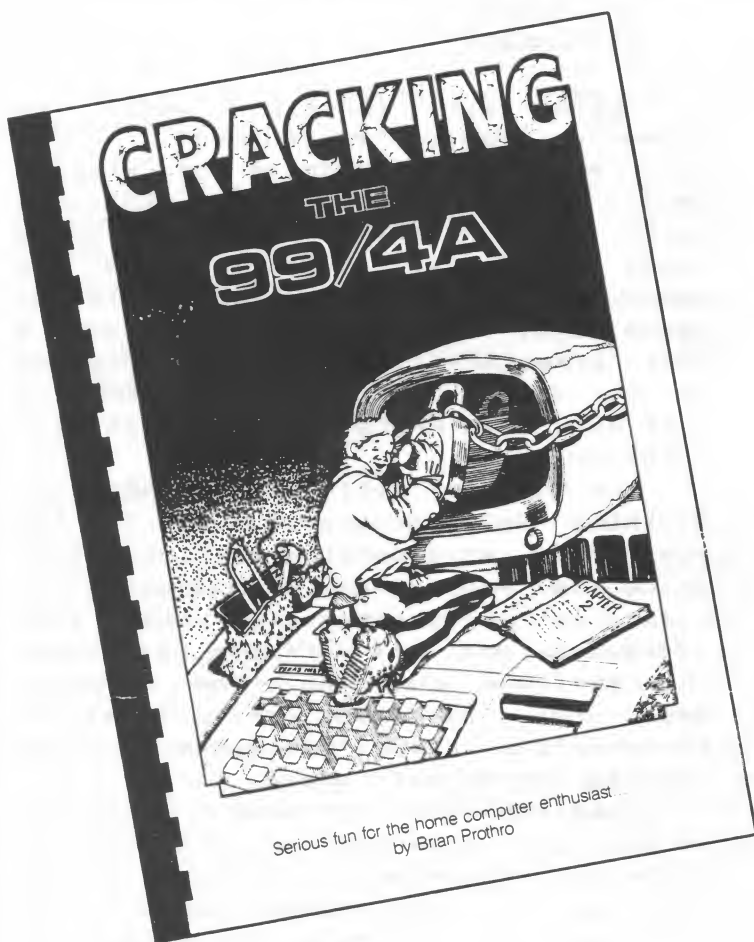
#include dsk1.random;c

```


CRACKING THE 99/4A

They said the 99/4A was a white elephant, they said it was expensive, they even said it was ugly! But it's still here, and all of us 99ers got the last laugh. So to put one laugh on top of another, from deep in the heart of Texas comes a new book. CRACKING THE 99/4A, written by Brian Prothro and illustrated by Wiley Akins.

CRACKING THE 99/4A is an intermediate level book. There are games and tips (as usual) and for many programs, the code is explained in detail. There are two tutorials. One is a very thorough lesson on Basic that covers structured programming, and linked lists (The sortless sort.) The second tutorial leads you through the powerful use of logical operators—from scratch.

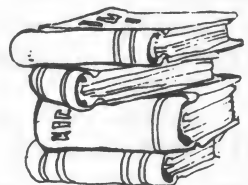


Would you like to use speech in your programs, but it's so much trouble you say "why bother!" This book includes two utilities that allow you to edit allophone strings and/or test speech slope parameters. (Tone of voice.) You'll find some graphics here and a few tips there. And for you dyed in the wool hackers, there are file I/O routines in assembly with additional code that lets you inspect a merged file.

All in all this is a uniquely different book. It is easy going and has tried to hit the middle of the road for all users. A full diskette is available which contains most of the major programs in the book. This should save you a lot of trouble, some of these programs are long.

You're interested—Right! So drop an order in the mail, and if not an order, send us a letter. Any questions can be directed through the address below.

	COVER PRICE
BOOK	\$12.95
DISKETTE	\$ 6.95
BOOK & DISKETTE ..	\$16.95



CRACKING THE 99/4A

MIDNIGHT EXPRESS PUBLISHERS

DEPT. UGL

P.O. BOX 26941

AUSTIN, TEXAS 78755

No C.O.D's

Add \$1 shipping per book. Maximum \$3 dollars.

supported 192k (128k main & 64k expansion) of video RAM, you can do page switching where a full 'page' of display is put on the screen in less than 1/30th of a second. When the display is shut off or idle, logical operations can be programmed using the on-chip logic. With fairly standard programming the chip can perform operations such as high speed move CPU to VRAM, VRAM to VRAM, high speed block moves, VDP to VRAM, Logical move CPU to VRAM, Logical VRAM to CPU etc. Other commands will search for border colours, draw lines and points in video or expansion ram.

Interfacing the 9938 chip to the TI 99/4A has proven to be a non-trivial task. In their design, TI laid out the video interface to accomodate users with TV sets or composite monitors, rather than allowing different video displays to be implimented.

Apparently a few other individuals have attempted to interface the 9938, usually on a PE Box card, to the 99/4A without success. The approach which has worked has been a combination of using the I/O port connectors and a connector to the 9918A video chip on the 4A motherboard.

This has the added advantage of being useful for all owners from console & cassette to full scale expansion systems.

The approach which we are licensing is designed to fit INSIDE the console without a stand-alone case on the side. This will prove to be easier and less expensive. The modifications required will be simple to accomplish for average owners. Those who do not wish to perform such modifications will be offered the upgrade service.

We would like to see letters from all of you discussing this! stating your preferences and demonstrating the viability of this idea!

BASIC COMPILER V 1.1

TI Basic is quite powerful. It is also slow. This is due to three factors: Basic is interpreted each and

every time you run a program, it is written in GPL; which is also interpreted inside the console AND it resides on GROM chips which are read sequentially a byte at a time. GROMs are a novel and intruiging, but they are best suited to sequential data like lists, arrays, screen tables etc.

Ryte Data has just started marketing a very straightforward Basic compiler which makes TI programs run much faster. The Compiler environment consists of a set of programs:

LOAD
COMPILER
FP-LOADER
INT-LOADER

These all function under an Extended Basic environment which requires 32k, a disk drive and Extended Basic. Upon selecting #2 Extended Basic from the main screen the program loads automatically into the 8k low memory. You are presented with the above menu. Choosing #1 gives you a disk directory listing of the programs on your disk. You can scroll down the list with FCTN PROCD or back again with FUNCTION REDO.

The Compiler will take a BASIC or EXTENDED BASIC program "as is" for the most part. It generates a 'code-list' which is stored as a program. This code list contains all the jump addresses and all variable addresses. As these are compiled, the computer does not need to re-interpret or re-calculate these addresses! giving you the increase in speed.

Once a program has been compiled it is stored on disk with an asterisk to differentiate between programs. If you have used strictly whole numbers (no fractions in values or variables) you can load and run the compiled program using the INTEGER loader. This is the faster of the two as FLOATING POINT uses certain console routines to calculate numbers.

There are two commands in XB which cannot be used: SUB for creating 'called' subroutines and DEF for defining calculated statements. The reasons for this are two fold: the Basic Compiler stops the process when

it reaches the END statement. Any subroutines used must be written in as program modules within the body of the program. Secondly, DEF statements are difficult to "unravel" under program control. Again you must write your program to compute a DEF statement as a usual command set.

Five other commands can cause strange results if used: CALL LOAD and CALL LINK due to the compiler residing in low memory; TRACE, ON ERROR and BREAK will not compile properly 100% of the time - it is best to avoid these statements. Otherwise the Basic Compiler fully supports all Basic and Extended Basic statements AND it will compile multi-statement lines in Extended Basic. You do not need to rewrite your programs or use single statement lines.

It is also possible to break a long program down into segments and use the RUN "DSKx.PROGRAMx" statement to link larger programs together.

Order a copy today! It's only \$20.00 (US) or \$28.00 (Cdn).

COVER DATE

You'll notice that the title screen says "August"... for good reason. As it takes a goodly amount of time and effort mixed with a liberal dose of time to publish a newsletter, we decided to reflect when you are likely to see the current issue. Many folks have asked "when this months issue" is coming out. So, THIS is this months issue. Your 1986 run still lasts to December - at which time you can make a resolution to renew.

While beating the drum, several treats are in store. For those who want more in-depth technical information we have an article on console design; an project for exchanging your 9918A Video Display Processor for a 9928 VDP (to get sharp monochrome & RGB output!), a diagram on altering your power supply, AND a review on the first working 80 Column Unit (at last!).

MYARC

The most recent issue of Micropendium had full page ads by Myarc for their Geneve 9640 Family Computer. It's nice to see Myarc start advertising their machine! Then again, if TI had been willing to license certain design & operating system software it could have been a lot sooner.

A great deal of talk has also centred around an 8088 co-processor to run MS-DOS / IBM software. There are two problems here. One is that a standard TI card is not large enough to hold a complete 8088 system. A co-processor board designed to fit a new PE Box (as per last issue) could work in this situation. This option would add another \$250 to the cost of the machine. Secondly, many users would simply buy an inexpensive clone rather than shelling out for a hybrid machine.

I can see how this approach would help Geneve move into business markets - perhaps in specific "vertical markets" - but this would have to rely on running IBM programs. Again, IBM clones would make more sense unless a very innovative screen handling (to use the advanced 9938 features / parallel processing/ elegant INPUT/ OUTPUT scheme was used.

Another aspect to consider, given the speed of Geneve and the video display, would be an "emulator" program written in assembly code. THIS could give the TI community IBM compatibility without the cost of a co-processor. With one (perhaps two) versions of "c" being written for Geneve, porting mainstream software to the new machine will be very easy. On the other side of that coin is the cost of top flight software!

We certainly plan on buying Myarcs new computer in the immediate future. We'll let you know what happens. If you are considering a more powerful machine - buy this one!

BILL GRONOS ?

We received a letter from Bill In Spain... specifically on a bus ride over (obviously) poor terrain to Madrid. Sounds like our article is delayed to next issue. His duties are demanding 16 hours a day to problem shoot some equipment problems.

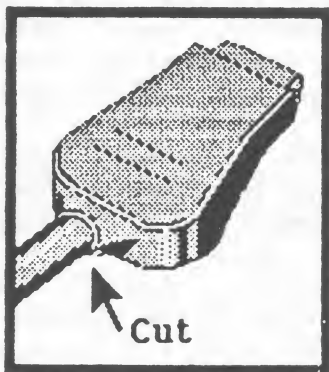
I suppose that, with the time lag overseas means we'll see some more great material, later.

Hold tight, he hasn't disappeared.

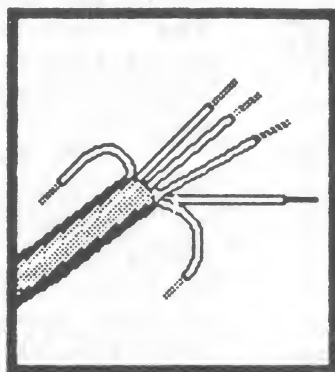
- by Mark G. Webb

Atari to TI-99/4a Joystick Conversion

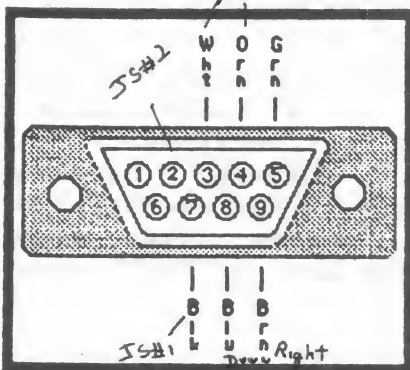
Here is a way to use that old joystick that has been under foot since your Atari game system was superceded by your TI.



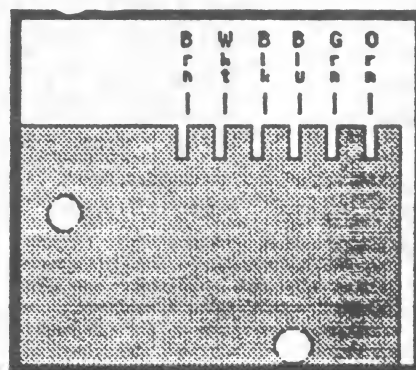
Cut the cable as close to the plug as possible to keep as much length as possible.



Trim back the outer insulation $\frac{3}{4}$ to $\frac{7}{8}$ of an inch. Then trim back the insulation on each wire $\frac{3}{16}$ of an inch.



Tin the wire tips and solder in the following manner:
 White to pin 3 Black to pin 7
 Orange to pin 4 Blue to pin 8
 Green to pin 5 Brown to pin 9



It is not necessary to open the joystick to complete the process but if you do these are the wire connections on the board inside.

INCENTIVE PRODUCT DRAW

(outright bribery time)

During the next 90 days, until November 21st, after the Chicago Users Group TI FAIRE, we are conducting a subscriber "incentive" promotion:

The product give away is open to all TI 99/4A owners - two classes of prizes will be offered. For entrants who send in their name and address on a postcard marked "99/4A INCENTIVE", we will be giving away the following products: your choice of-

1. Navarone 'Widget' Cartridge Expander
2. StarShip Pegasus by Not-Polyoptics (cassette Basic)
3. Night Mission by Millers Graphics (cassette Extended Basic)
4. Microsurgeon (TI module)

For subscribers to R/D COMPUTING the draw will include your choice of:

1. Extended Basic II plus (high resolution module)
2. 32k Memory Expansion unit
3. GPL Assembler Package with INTERN and GPL Linker (32k/disk)
4. MiniMemory Programming Module
5. Super Clock Support, INTERN book and Basic Compiler (disk)
6. Lifetime subscription w/ disk to R/D COMPUTING.

Something for everyone! No purchase is required to enter the draw. Send in your name and address on a postcard, stating your choice of product. The draw is open to all TI 99/4A owners, subscribers and users groups. **REMEMBER**, our promotional draw closes November 25th, 1986. Winners names will be published in the December 1986 issue. Please pass the word along to your friends, your user group and any other 99 owners!

DO IT TODAY!

CONFESSIONS OF AN EX-PIRATE anonymous

Can you remember way back, when you first bought your computer system? You spent a lot of your hard earned money on all these real neat (and expensive) pieces of hardware. And it was great! Lights came on, things made noise, disk drives grunched... but something was missing... SOFTWARE!

No big deal. Just run down to the store where you bought your system and spend some more money... "Whatin the %(%!! doyoumean \$99.95 for TI WRITER?."

"Multiplan and Extended Basic cost that much too???"

"#%\$*!#+(!!"

"What do you have for under \$40.00? Munchman, Parsec, Blasto and Teach Yourself Extended Basic? You have got to be kidding!!!"

So out you go and make some friends. One of them hands you a disk saying "Have I got something for you - don't pass this on to anyone else."

You run home, shove the disk into your machine, and you catalog it...

"Hmmm, this isn't too bad; a couple of public domain BASIC programs, a user written checkbook balancer, and... Whats this BUCKROGERS? How did that guy get this?"

So, quickly making a backup copy, you return the "original" to your "friend" And then you say the magic words... "GOT ANYTHING ELSE?" The next time your friend gives you five disks. Again you rush home to your computer and fire it up.

"Boy, this stuff is great! Arcade games, Text games, Productivity software..."

A friend stops by your house complaining about the high price of software... And you say, "No problem! Have I got something for you! Waddaya want? I've got everything."

Sound familiar? Most of us (myself included) at some time or another have done this to some degree. Some make it their mission in life. After, all, what is the harm in sharing a program with a "friend"?

The computer industry name for this

practice -PIRACY- sounds kind of daring and exciting.

Why don't we call it what it really is --- THIEVERY!!! Stealing software is no different than walking into a store, taking an item off the shelf, and walking out without paying for it. Except that it is harder for the software author to get you for stealing his or her product.

If programmers wanted to give their programs away, they wouldn't have put price tags on them!!!

There are a few more backlashes from the act of piracy (stealing). Copy protected software and lack of new product development.

Let us first look at copy protection. Personally, I would rather a software developer to spend more time on finding ways to get more out of my machine than how to guard his or her latest masterpiece. Just so they can make a few well-earned bucks.

Most protection schemes are either broken or bypassed within a week or two of release. Protection schemes also make your drives work harder (listen when a protected program is loading - the heads are really moving back and forth across the disk) resulting in more wear and tear on your equipment. Sounds great! Some disk controller cards will work just fine on non-protected programs but they won't even run protected software!!!

How about making back-ups of software you did purchase? Forget it! For some real fun, try installing protected software on a hard disk or a RAM disk. You can't! Or trash your original (and only) program disk. You need that mail list or report finished by the end of the week? Sorry, 3 to 6 week wait. Plus you must send in your original disk. Pity the poor author who doesn't copy protect. Well, he or she will get famous, not rich, just famous. These are just a few of the benefits of STEALING software.

Does anyone realize just how important new product development is to continued existence of our TI's? Just in case no one told you, Texas Instruments doesn't supply 99/4A hardware or software any longer. Who does? It is the small companies. There

ADVANCED DIAGNOSTICS

ADVANCED DIAGNOSTICS				
Status				
Command : Format				
Drive	: 2	Track	: 0	
Side	: 1	Sector	: 1	
Free	: 700	H/S	: 3	
Used	: 20	Bad	: 18	
Sectors Mapped Bad				
190	208	215	226	233
244	251	262	269	280
287	298	305	316	323
334	341	352		
....				
....				
....				
....				
[Edit Sector 251]				
[]				
Disk Initialization Completed				

ADVANCED DIAGNOSTICS				
Status				
Command : Motor Speed				
Drive	: 2	Track	: 0	
Side	: 1	Sector	: 0	
Free	: 700	H/S	: 3	
Used	: 20	Bad	: 18	
Motor Speed (RPM)				
280 290 300 310 320				
.....				
[Motor Speed]				
[]				

Some of the more advanced features include Windowed HELP at any time which displays all of the full command names, their abbreviation and their syntax, and the FCTN and CTRL key assignments. It also gives you the ability to execute user written Command Files and a Screen dump function with the press of a key. Command Files are DIS/VAR 80 Text files like TI Writer and Editor/Assembler files. These Files can contain any of the commands known to Advanced Diagnostics and any of the keys on the keyboard (ie: FCTN 1 Delete etc.). Once the command file has been loaded into memory Advanced Diagnostics will execute it as if it was typed in from the keyboard, only MUCH faster!! With this feature you can easily write your own Command Files to: Print out selected sectors from the disk to your printer or Edit sectors and change selected characters and hex values or generate some protection or.....

Advanced Diagnostics comes with 7 Command Files that preform the following tasks: Diagconfig - to configure Advanced Diags to your system, Boxformat - to format a box of ten diskettes, Dskcopy1-2 - to format the disk in drive 2 and then copy the disk in drive 1 to drive 2, Diskheader - to show you how to use Edit Sector in a Command File, R&Wtest - to do a Read and Write test on the selected drive, Readtest - to do a read only test on the selected drive and Seektest - to do a seek track test on the selected drive.

Here are the Programmable ADVANCED DIAGNOSTICS Commands

BEEP	CONVERT NUMBER	FORMAT DISK	OUTPUT WIDTH	TIME DELAY
CHANGE COLORS	COPY READ	FIND FILE	PAUSE	USE DSR
CHECK DISK	COPY WRITE	HEAD STEP	READ TRACK	WRITE SECTOR
COMMAND FILE	DISK DIRECTORY	MOTOR SPEED	SEEK TRACK	WRITE TRACK
CHECK MEMORY	EDIT SECTOR	OUTPUT DEVICE	SELECT DRIVE	

This new program comes with 40 pages of documentation that also includes an introduction into the theory of operation of a floppy disk controllers and complete memory maps of the important sectors on a diskette. This program is fully compatible with the TI disk controller, Myarc's 128K RAM Disk and the corcomp disk controllers. Requires: TI 99/4A, Memory Expansion, Disk System and one of the following modules - Extended Basic, Editor/Assembler or Mini Memory. (currently not compatible with the Myarc disk controllers)

MILLERS GRAPHICS

1475 W. Cypress Ave.
San Dimas, CA 91773
U.S.A. EARTH

ORDER NO. UT01

PRICE 19.95



is only one large software house writing for the TI and they write adventure games (seen any copies of THEIR stuff?). There are no Ashton-Tate's, Microsoft's or Lotus' writing for our beloved TI! Most of our software is produced by small companies. Do you really think that thy will continue to bring out new programs that will just go from one pirate to the next?

And how about those Freeware authors? Those people who work or go to school and write software at night and on weekends, sometimes for months, and then have the gall to ask you to send them ten or fifteen dollars if you like or can use their program. How many people ever do? According to these authors, not very many. The crazy part is that most of these programs are really quite good! Some like PRBase and DM1000 v3.1 are very useful. Those like Danny Michael's "Screen Dump" and "Neatlist" allow us to do things that we couldn't do before.

But send these people money? "Why should I, when I can get it for free?" What the pirate (thief) doesn't realize is that nothing is "FREE". As the man says, you can pay me now or pay me later. We ALL will pay for this "FREE" software by eventually having a machine that will only run the oldest, out-of-date programs around.

How long do you think they are going to write for this machine? Why the hell should they? So they can get ripped off some more? I don't think so.

Did you know that there is a TI BBS in our area "up and running" on PIRATED software? Do you think that the authors, most of whom write good software, are too stupid to know when they are getting screwed?

I predict that the future of our machine is very cloudy indeed. It's a simple formula: SOFTWARE + HARDWARE = PRODUCTIVITY AND/OR ENJOYMENT!!!

Then what are we going to do?

Well, to people that accept pirated (stolen) software, DON'T.

To those that do the pirating (stealing), KNOCK IT OFF!!

To software producers, please charge a reasonable price. And if you must protect your goods, please put two copies in the package so that the

legitimate users are protected too.

I sucked up software with the best of 'em. No more!!! Why? I saw that, for machines that are in production, there are 15 to 30 titles PER MONTH being released for them. But for orphan computers - zip. So far we've been lucky. Are you willing to push that luck? I'm not.

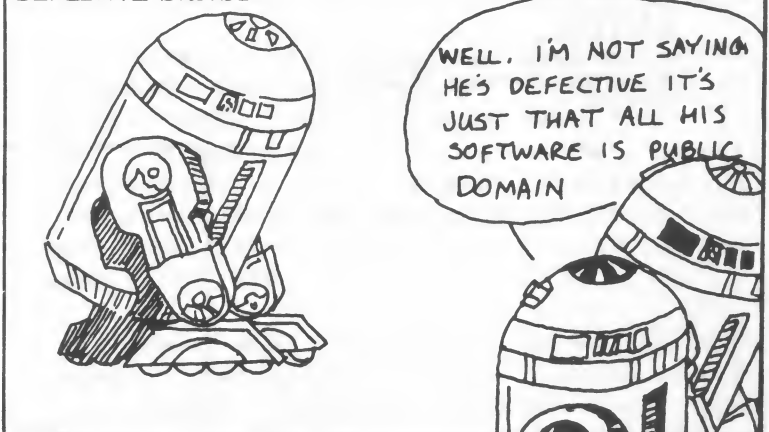
taken from the Puget Sound 99ers.

TIGERCUB SOFTWARE

Over 130 original entertainment, education and programming utility programs in Basic and Extended Basic on cassette or disk. Only \$3.00 each! Eighteen different full-disk collections, just \$12.00 each! Descriptive catalogue \$1.00 refundable. TIPS FROM TIGERCUB full-disk collections of 50+ programs and files from Tigercub Tips newsletters, Vol. I, II and III \$15.00 each! any two \$27.00! all three \$35.00 postpaid. NUTS & BOLTS (#1) and (#2) full disks of 100+ utility subroutines in XBasic 'merge' format, ready to merge into your own programs: \$19.95 each, both for \$37.00 with documentation, postpaid. Orders to Tigercub Software, 156 Collingwood Ave., Columbus, OH 43213

MAIL-LIST stores up to 400 records. Sort by name. Search by part of last name. Print labels & listings. Entire program loaded into memory eliminating numerous disk changes. Requires TI-99/4A, XBasic, disk drive, 32k mem. Printer optional. Write: K.G. Mori, 181 Atlantic Avenue, Winnipeg, Manitoba, Canada R2W 0P7

DEFECTIVE DROIDS



SUSCRIBE NOW to receive each new issue. Copies are mailed during the fourth week every month to subscribers. The data shown on each cover is for the month PAST ie: JAN 1986 is published at the end of January. Back issues are available as follows:

- V 1.2 Load Interrupt Switch
E/A 8k module upgrade
- V 1.3 TMS 9995 Memory map & specs
numeric keypad project & etc.
- V 1.4 Extended Basic plus by Apesoft
Myarc 128k card
Auto-fire project & etc.
- V 1.5 32K internal memory upgrades
DS/DD Ramdisk
Auto power-up project
- V 1.6 Myarc 256k Computer
"C" Compiler
128k 'console only' memory unit
RAM/GRAM card
- V 1.7 MAXIMEM review
EPROM burner
Schedule Manager review
Sense & Control card

1986 Subscription price: \$14.00 US funds including First Class Delivery. Add \$3 foreign

Back Issues are available to subscribers only. Price: \$2.00 each. Add \$1.00 postage US & Canada. Add \$2.00 overseas. For Special Delivery AIRMAIL add \$4.00.

R/D Computing is published monthly by RYTE Data in Haliburton, Ontario. Copyright 1985. All material contained herein is taken from sources believed to be accurate. No responsibility for errors, omissions or misprints is assumed. Articles may be reprinted with credits by users groups for publication.

Articles dealing with Texas Instruments 99/4A and 9900 based computers are published. Special attention is given to data on upgrading and modifying the 4A console & system. Information and material for consideration is solicited from owners, users groups; hardware manufacturers, software publishers etc. Please send all correspondence to RYTE Data R/D, 210 Mountain Street, Haliburton, Ont. K0M 1S0 Canada. We cannot accept responsibility for materials submitted and, unless stated otherwise, will assign all manuscripts, letters etc. for publication. **ONLY** Manuscripts with sufficient postage and self-addressed mailer will be returned.

In two issues (March) we are setting up a new un-classified ad section. Cheap rates at \$1.25 per 40 character line. Send copy to our address with payment. Count all the characters, spaces etc. Ads will be run according to order received. Deadline date is the 21st of each month.

.....Rytetm
Data.....

BOX 210 MOUNTAIN STREET
HALIBURTON, ONTARIO K0M 1S0
CANADA

